US009373073B2

(12) **United States Patent**
Arthur et al.

(10) **Patent No.:** **US 9,373,073 B2**
(45) **Date of Patent:** ***Jun. 21, 2016**

(54) **TIME-DIVISION MULTIPLEXED NEUROSYNAPTIC MODULE WITH IMPLICIT MEMORY ADDRESSING FOR IMPLEMENTING A UNIVERSAL SUBSTRATE OF ADAPTATION**

(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION**, Armonk, NY (US)

(72) Inventors: **John V. Arthur**, Mountain View, CA (US); **Bernard V. Brezzo**, Somers, NY (US); **Leland Chang**, New York, NY (US); **Daniel J. Friedman**, Sleepy Hollow, NY (US); **Paul A. Merolla**, Palo Alto, CA (US); **Dharmendra S. Modha**, San Jose, CA (US); **Robert K. Montoye**, Rochester, MN (US); **Jae-sun Seo**, White Plains, NY (US); **Jose A. Tierno**, Stamford, CT (US)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 309 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **13/725,463**

(22) Filed: **Dec. 21, 2012**

(65) **Prior Publication Data**

US 2014/0180984 A1 Jun. 26, 2014

(51) **Int. Cl.**

G06E 1/00	(2006.01)
G06E 3/00	(2006.01)
G06F 15/18	(2006.01)
G06G 7/00	(2006.01)
G06N 3/02	(2006.01)

(Continued)

(52) **U.S. Cl.**
CPC ***G06N 3/02*** (2013.01); ***G06N 3/049*** (2013.01); ***G06N 3/063*** (2013.01); *G06N 3/0454* (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,994,982	A	2/1991	Duranton et al.
7,908,235	B2	3/2011	Snook et al.
8,126,828	B2	2/2012	Snook et al.

(Continued)

OTHER PUBLICATIONS

Large Developing Axonal Arbors Using a Distributed and Locally-Reprogrammable Address-Event Receiver Simeon A. Bamford, Alan F. Murray, David J. Willshaw 978-1-4244-1821-3/08/$25.00 c 2008 IEEE.*

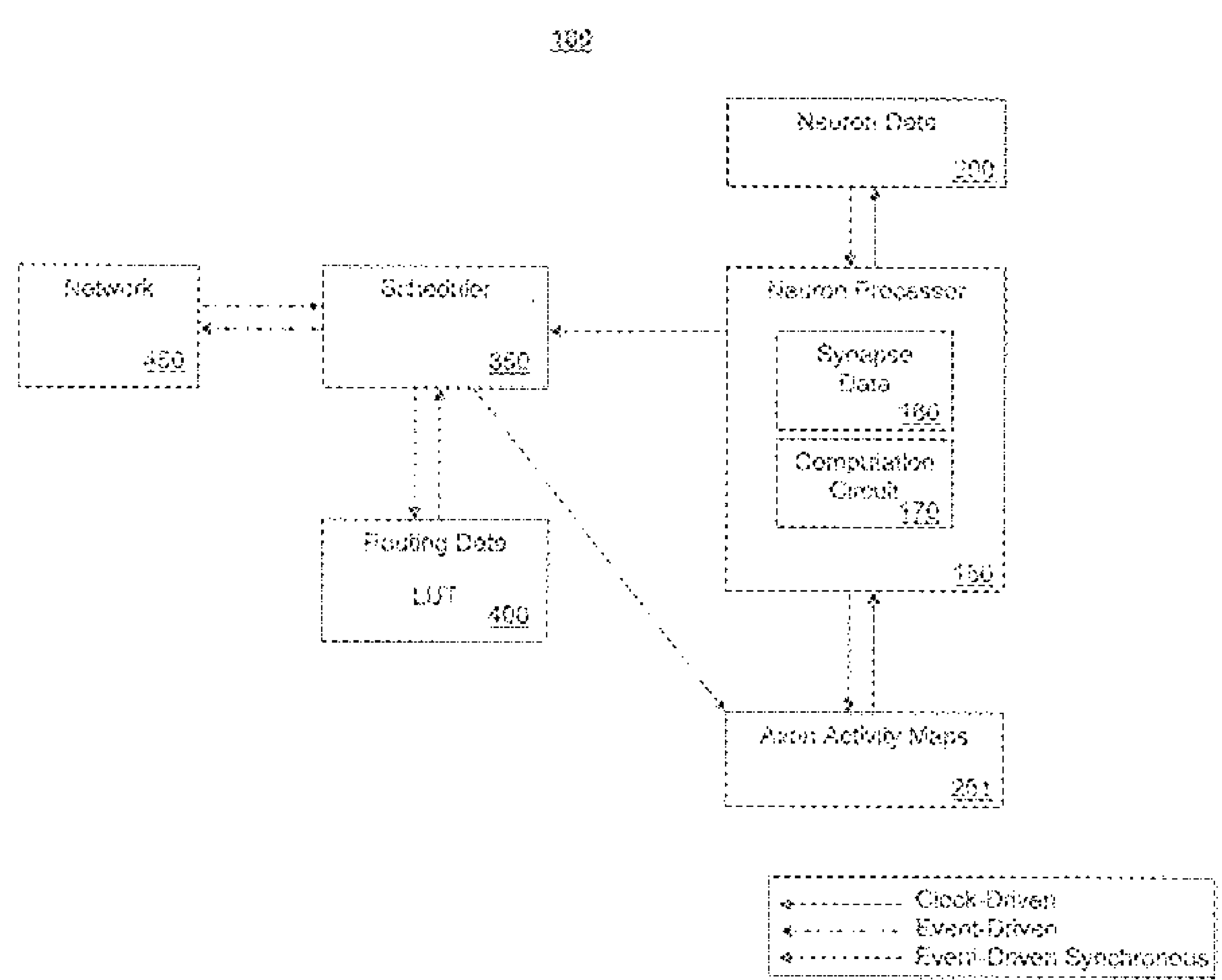
(Continued)

Primary Examiner — Kakali Chaki
Assistant Examiner — Ababacar Seck
(74) *Attorney, Agent, or Firm* — Sherman IP LLP; Kenneth L. Sherman; Hemavathy Perumal

(57) **ABSTRACT**

Embodiments of the invention relate to a time-division multiplexed neurosynaptic module with implicit memory addressing for implementing a universal substrate of adaptation. One embodiment comprises a neurosynaptic device including a memory device that maintains neuron attributes for multiple neurons. The module further includes multiple bit maps that maintain incoming firing events for different periods of delay and a multi-way processor. The processor includes a memory array that maintains a plurality of synaptic weights. The processor integrates incoming firing events in a time-division multiplexing manner. Incoming firing events are integrated based on the neuron attributes and the synaptic weights maintained.

31 Claims, 10 Drawing Sheets



(51) **Int. Cl.**
G06N 3/063 (2006.01)
G06N 3/04 (2006.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,326,782	B2	12/2012	Snook et al.	
8,423,490	B2	4/2013	Deco et al.	
8,510,239	B2	8/2013	Modha	
8,725,662	B2	5/2014	Izhikevich et al.	
8,856,055	B2	10/2014	Brezzo et al.	
2007/0288410	A1*	12/2007	Tomkins et al.	706/42
2010/0241601	A1*	9/2010	Carson	G06N 3/02 706/27
2010/0312735	A1	12/2010	Knoblauch	
2011/0051485	A1*	3/2011	Chang	G11C 15/00 365/49.17
2012/0084240	A1	4/2012	Esser et al.	
2012/0109866	A1*	5/2012	Modha	G06N 3/063 706/28
2013/0031040	A1*	1/2013	Modha	G06N 3/049 706/27
2013/0073491	A1	3/2013	Izhikevich et al.	
2013/0073497	A1*	3/2013	Akopyan	G06N 3/04 706/27
2014/0180987	A1	6/2014	Arthur et al.	
2014/0222740	A1*	8/2014	Alvarez-Icaza Rivera	G06N 3/063 706/28
2014/0344201	A1*	11/2014	Arthur	G06N 3/02 706/25

OTHER PUBLICATIONS

Learning in Silicon: A Neuromorphic Model of the Hippocampus John Vernon Arthur A Dissertation in Bioengineering—2006.*

Synaptic Rewiring in Neuromorphic VLSI for Topographic Map Formation Simeon A. Bamford The UniversityofEdinburgh Doctor of Philosophy Institute for Adaptive and Neural Computation School of Informatics University of Edinburgh 2009.*

Scalable Event Routing in Hierarchical Neural Array Architecture with Global Synaptic Connectivity Siddharth Joshi , Steve Deiss†‡, Mike Arnold‡§, Jongkil Park , Theodore Yu ‡, Gert Cauwenberghs†‡ 978-1-4244-6678-8/10/$26.00 © 2010 IEEE.*

Dynamically Reconfigurable Silicon Array of Spiking Neurons With Conductance-Based Synapses R. Jacob Vogelstein, Student Member, IEEE, Udayan Mallik, Joshua T. Vogelstein, and Gert Cauwenberghs, Senior Member, IEEE.*

Merolla, P. et al., "A recurrent model of orientation maps with simple and complex cells", Department of Bioengineering, Dec. 9, 2003, pp. 1-10, University of Pennsylvania, United States.

Brette, R. et al., "Simulation of networks of spiking neurons: A review of tools and strategies", J Comput Neurosci, 2007, pp. 349-398, Springer Science, United States.

Marian, I. et al., "Efficient even-driven simulation of spiking neural networks", Proceedings of 3rd International Conference on Neural Networks and Applications, Feb. 2002, pp. 1-9, Interlaken, Switzerland.

Nageswaran, J.M. et al., "Efficient Simulation of Large-Scale Spiking Neural Networks Using CUDA Graphics Processors", Proceedings of the International Joint Conference on Neural Networks, Jun. 14-19, 2009, p. 1-8, IEEE, United States.

Choi, T. et al., "Neuromorphic Implementation of Orientation Hypercolumns", Proceedings of the IEEE Transactions on Circuits and Systems, Jun. 2005, pp. 1049-1060, vol. 52, No. 6, United States.

Merolla, P.A. et al., "Expandable Networks for Neuromorphic Chips," IEEE Transactions on Circuits and Systems—I: Regular Papers, Feb. 2007, pp. 301-311, vol. 54, No. 2, IEEE, United States.

Rossello, J.L. et al., "Spiking Neural Network Self-configuration for Temporal Pattern Recognition Analysis", LNCS 5768, ICANN 2009, pp. 421-428, Springer-Verlag, Berlin Heidelberg.

Seo, J. et al., "A 45nm CMOS Neuromorphic Chip with a Scalable Architecture for Learning in Networks of Spiking Neurons", Proceedings of the 2011 IEEE Custom Integrated Circuits Conference (CICC), Sep. 19-21, 2011, pp. 1-4, IEEE, United States.

Ananthanarayanan, R. et al., "Anatomy of a Cortical Simulator," Proceedings of the 2007 ACM/IEEE Conference on Supercomputing (SC '07), 2007, 12 pages, ACM, United States.

Harkin, J. et al., "A Reconfigurable and Biologically Inspired Paradigm for Computation Using Network-On-Chip and Spiking Neural Networks", International Journal of Reconfigurable Computing, 2009, pp. 1-14, United States.

Ananthanarayanan, R. et al., "The Cat is Out of the Bag: Cortical Simulations with 109 Neurons, 1013 Synapses", Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis SC'09, Nov. 14-20, 2009, pp. 1-12, No. 63, ACM, United States.

Glackin, B. et al., "A Novel Approach for the Implementation of Large Scale Spiking Neural Networks on FPGA Hardware", Proceedings of the 8th International Conference on Artificial Neural Networks: Computational Intelligence and Bioinspired Systems (IWANN '05), Jun. 2005, pp. 552-563, Springer-Verlag Berlin, Heidelberg, Germany.

Lin, J. et al., "Programmable Connections in Neuromorphic Grids", Proceedings of the 49th International Midwest Symposium on Circuits and Systems, Aug. 6-9, 2006, pp. 80-84, IEEE, United States.

Jin, X. et al., "Modeling Spiking Neural Networks on SpiNNaker", Computing in Science & Engineering, Sep.-Oct. 2010, pp. 91-97, IEEE, United States.

Harkin, J. et al., "Reconfigurable Platforms and the Challenges for Large-Scale Implementations of Spiking Neural Networks", Proceedings of the International Conference on Field Programmable Logic and Applications (FPL 2008), Sep. 8-10, 2008, pp. 483-486, IEEE, United States.

U.S. Non-Final Office Action for U.S. Appl. No. 13/725,476 mailed Jan. 6, 2015.

U.S. Final Office Action for U.S. Appl. No. 13/725,476 mailed Mar. 26, 2015.

U.S. Advisory Action for U.S. Appl. No. 13/725,476 mailed May 12, 2015.

Rivas-Perez, M. et al., "Visual Spike Processing based on Cellular Automaton", Cellular Automata—Simplicity Behind Complexity, Apr. 11, 2011, pp. 529-547, United States.

Rivas-Perez, M. et al., "Visual Spiked-based Convolution Processing with a Cellular Automata Architecture", Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Jul. 18-23, 2010, pp. 1-7, IEEE, United States.

U.S. Non-Final Office Action for U.S. Appl. No. 13/725,476 mailed Jul. 13, 2015.

U.S. Notice of Allowance for U.S. Appl. No. 13/725,476 mailed Sep. 9, 2015.

* cited by examiner

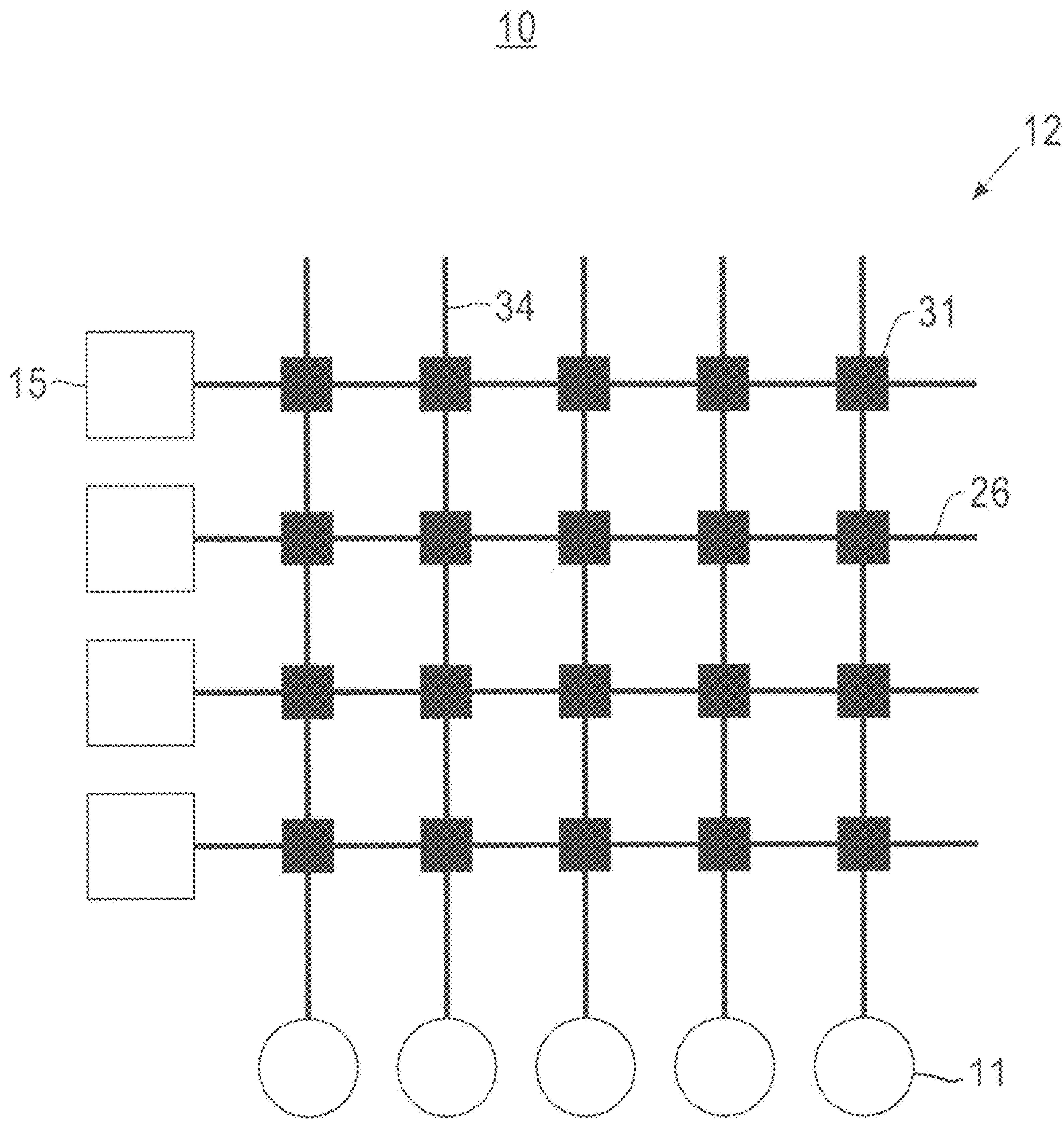


FIG. 1A

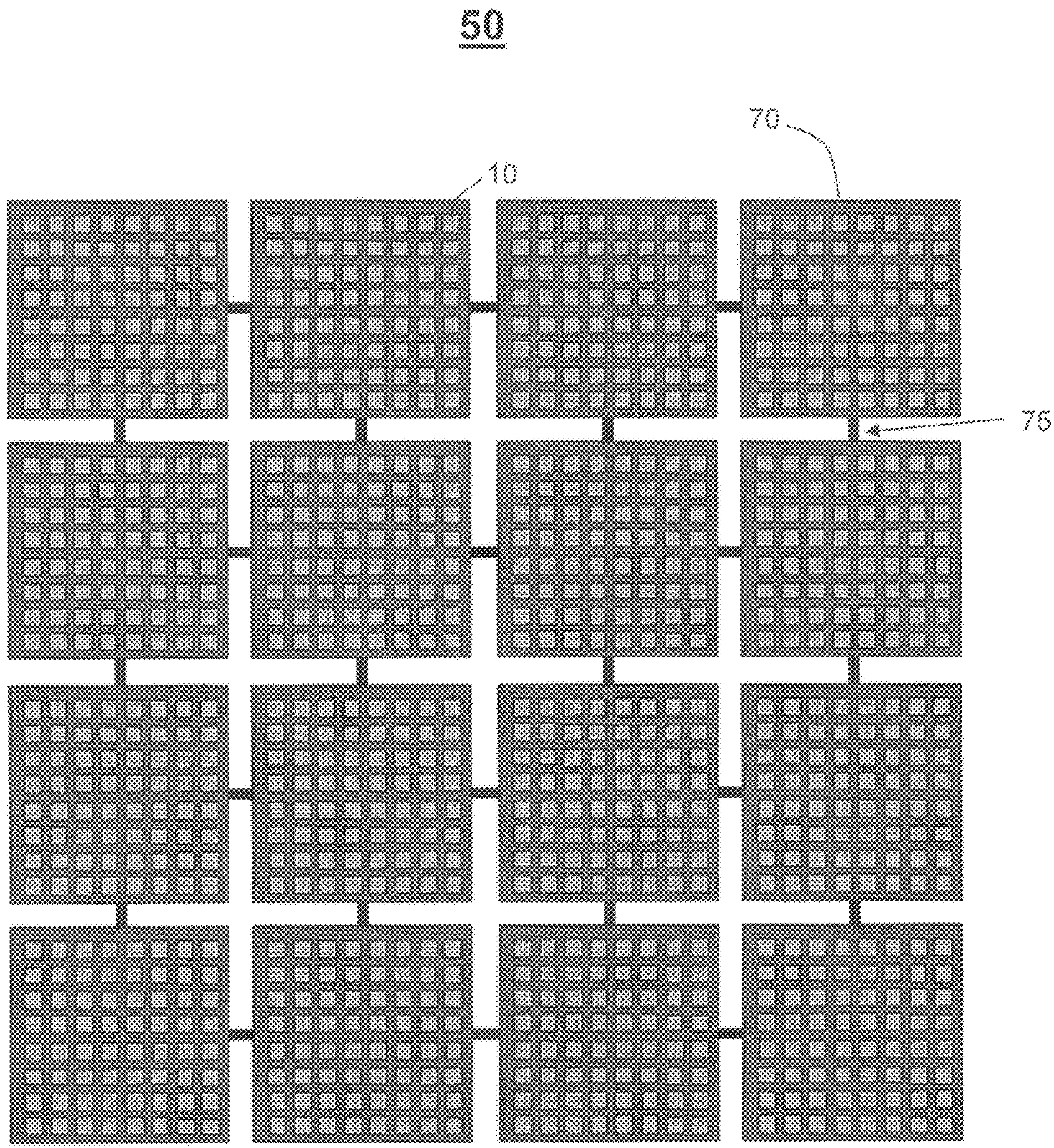


FIG. 1B

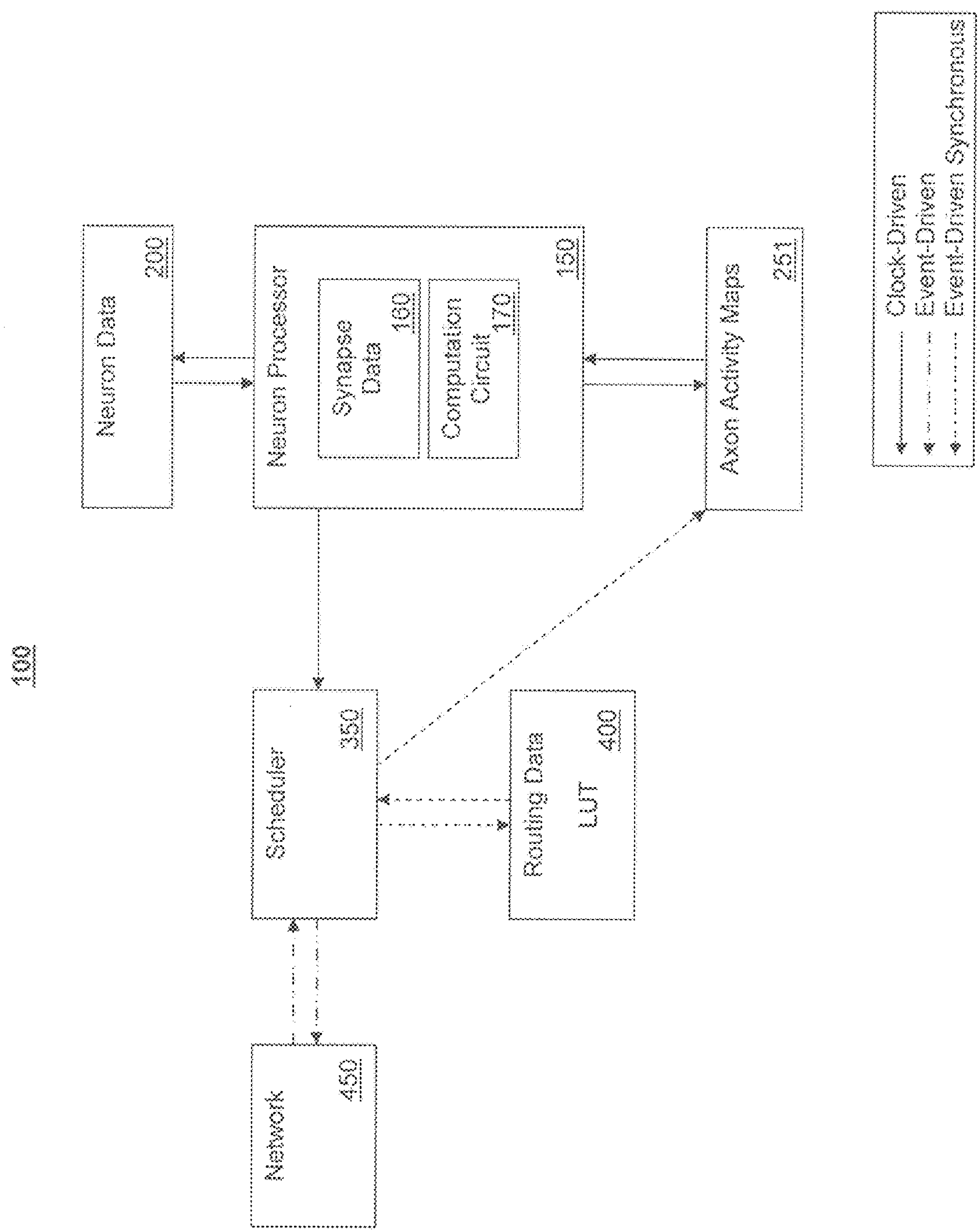

100
Neuron Data
200
Neuron Processor
Synapse Data
160
Computation Circuit
170
150
Axon Activity Maps
251
Scheduler
350
Routing Data
LUT
400
Network
450
Clock-Driven
Event-Driven
Event-Driven Synchronous

FIG. 2

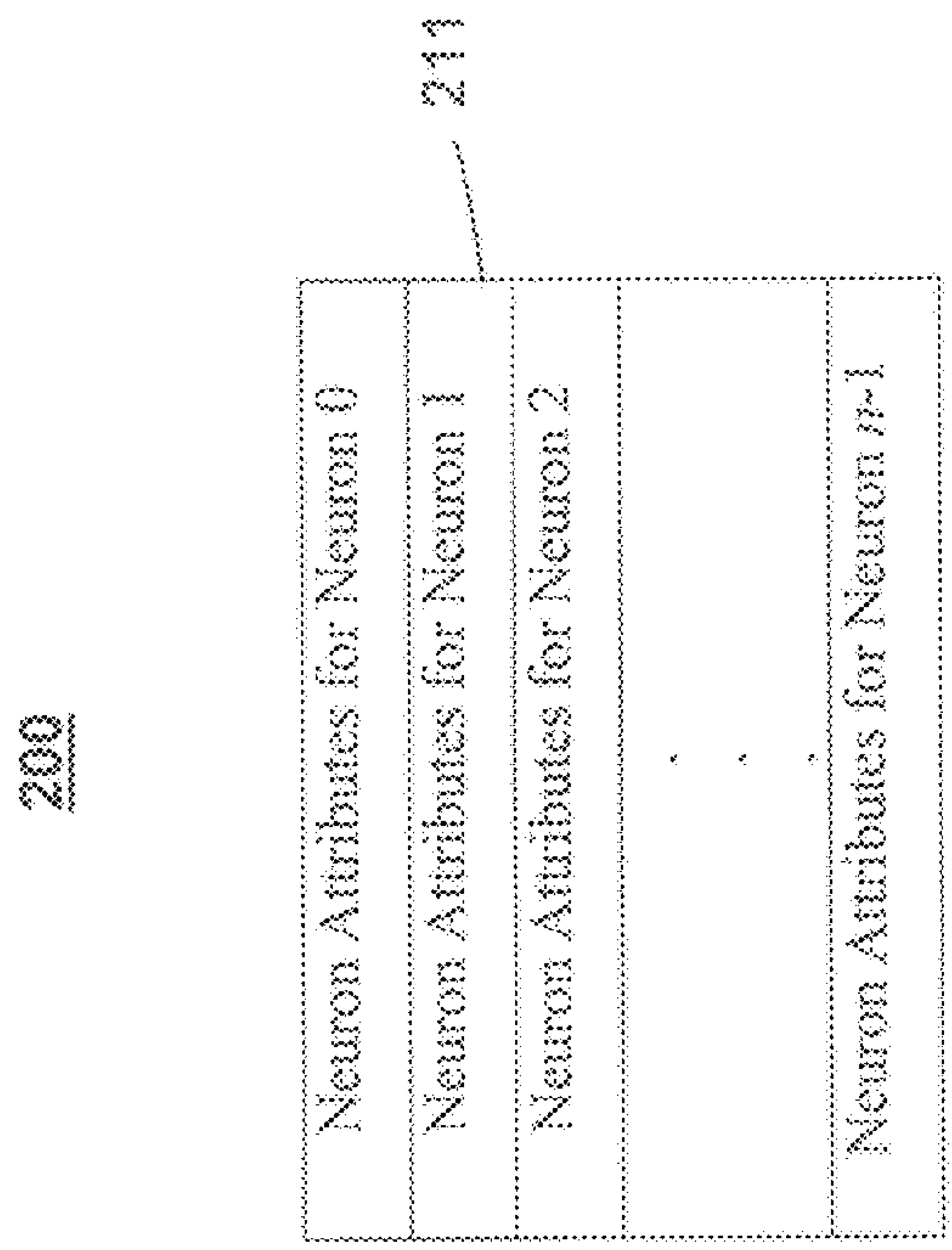

200
211
Neuron Attributes for Neuron 0
Neuron Attributes for Neuron 1
Neuron Attributes for Neuron 2
. . .
Neuron Attributes for Neuron n-1

FIG. 3A

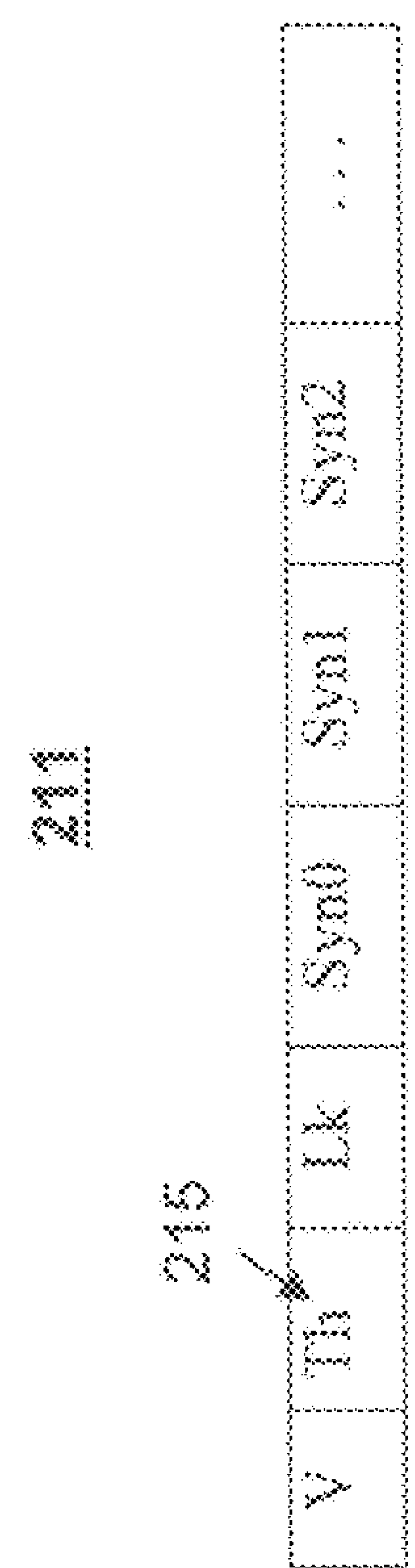

211
215
V
Th
Lk
Syn0
Syn1
Syn2
. . .

FIG. 3B

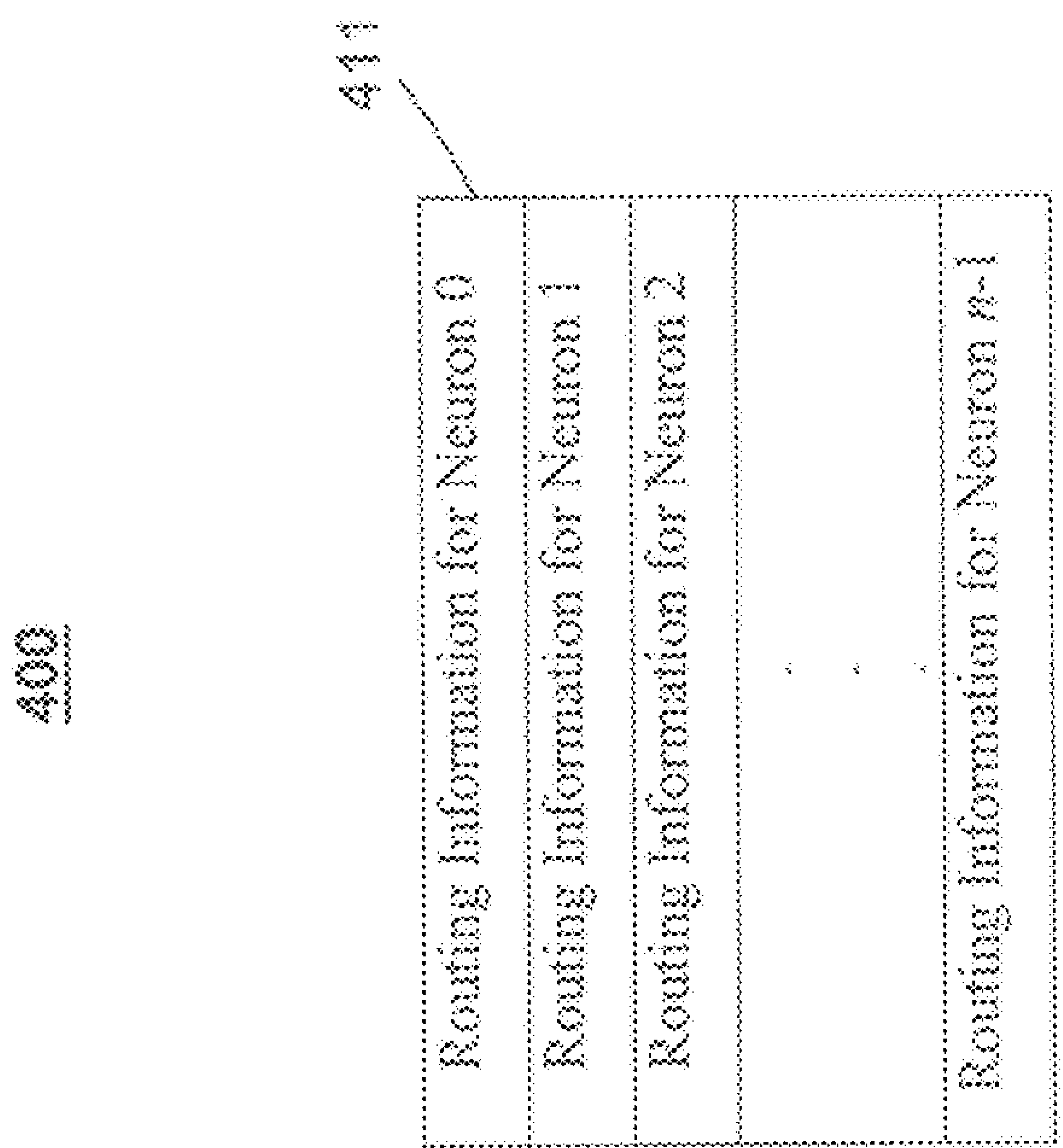

400
411
Routing Information for Neuron 0
Routing Information for Neuron 1
Routing Information for Neuron 2
. . .
Routing Information for Neuron n-1

FIG. 4A

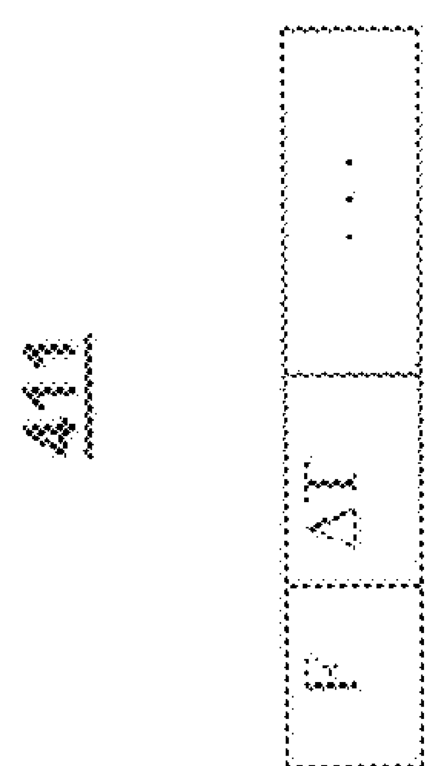

411
F
ΔT
. . .

FIG. 4B

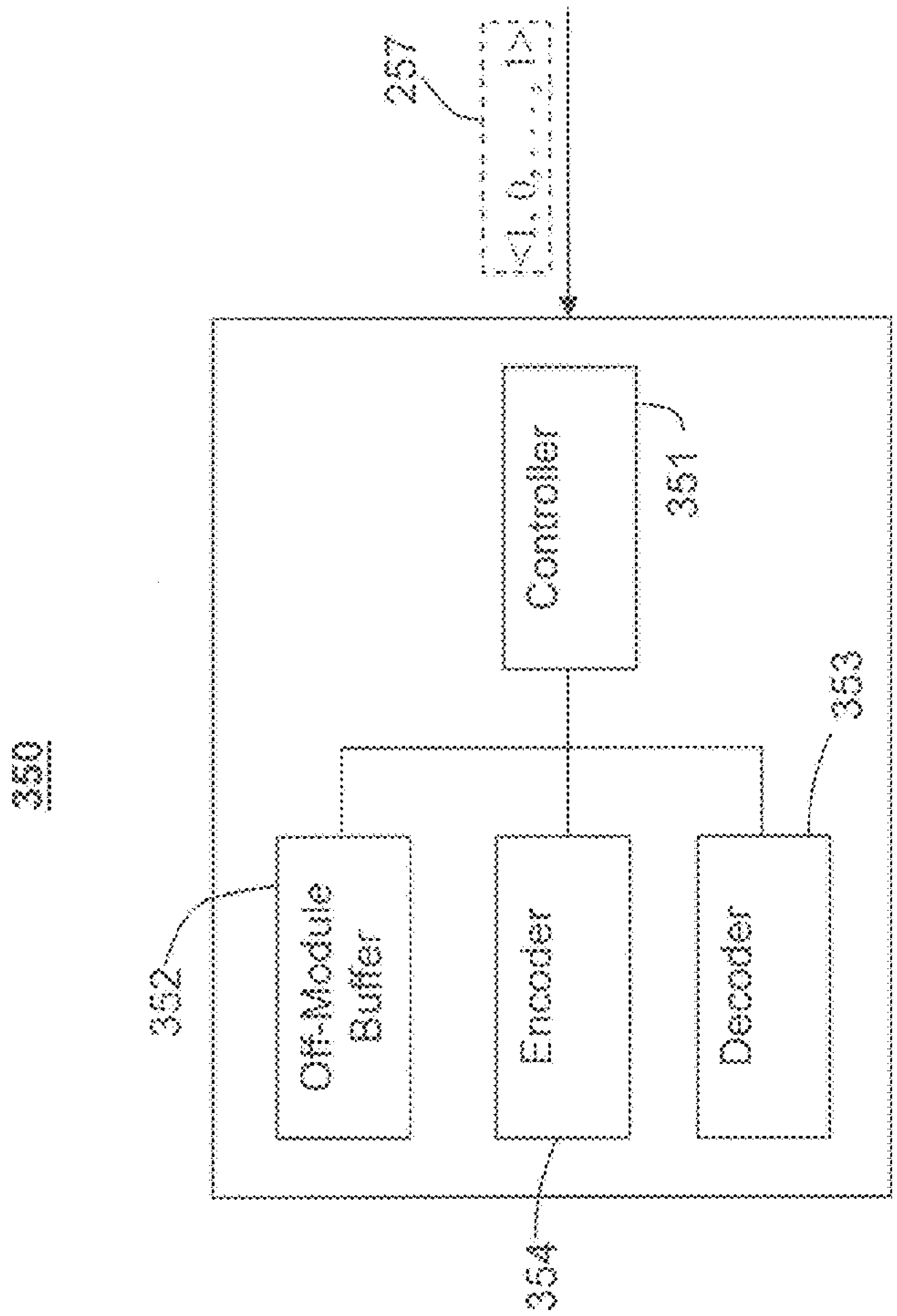

350
Off-Module Buffer
Encoder
Decoder
Controller
351
352
353
354
257
<1, 0, . . . , 1>

FIG. 5

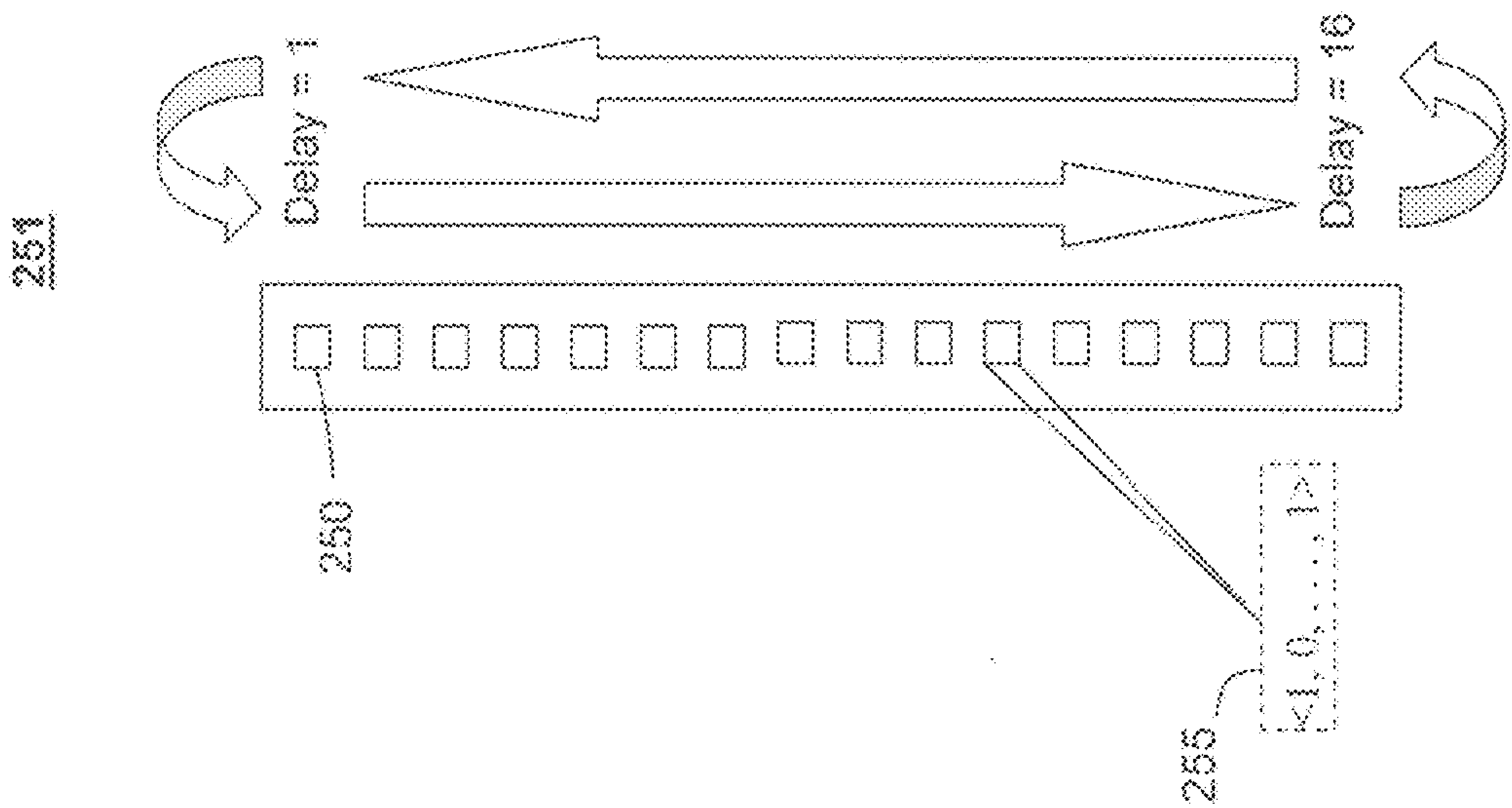

251
Delay = 1
Delay = 16
250
255
<1, 0, . . . , 1>

FIG. 6

160

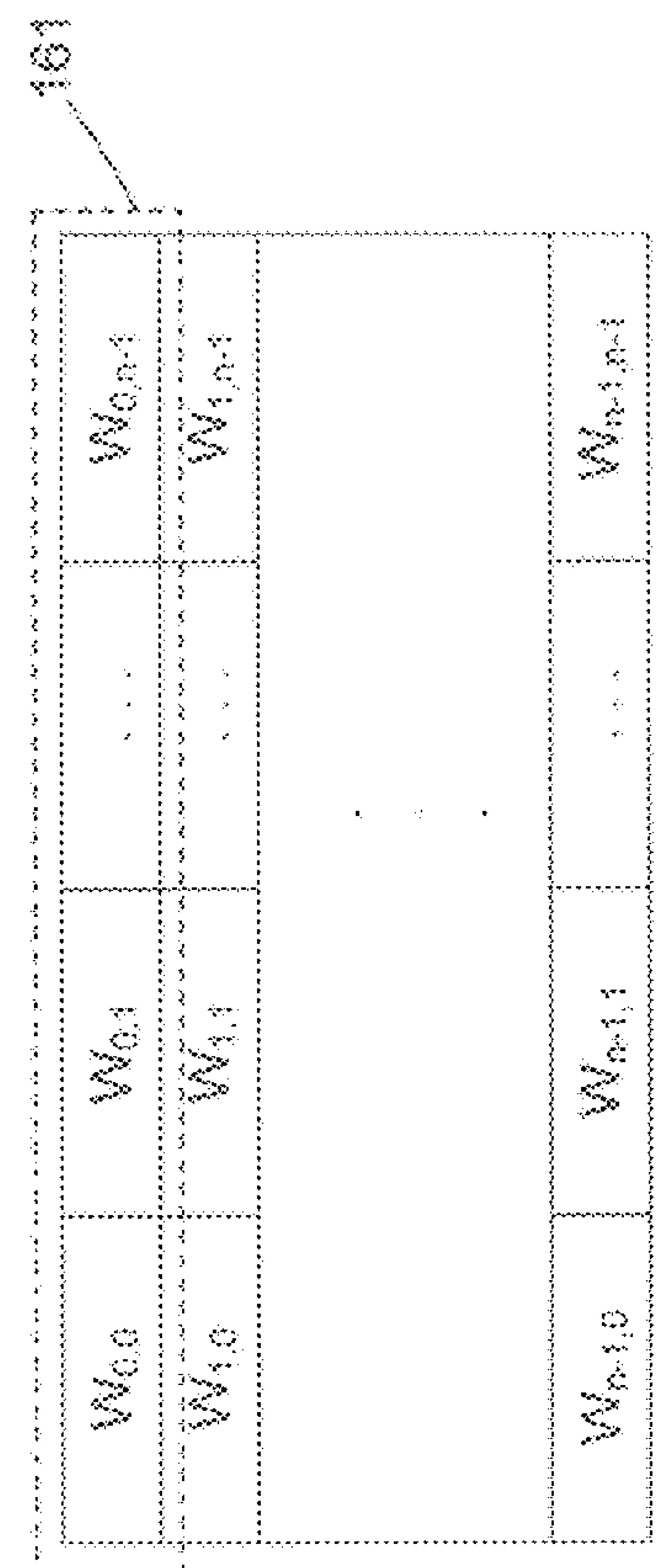


FIG. 7A

160

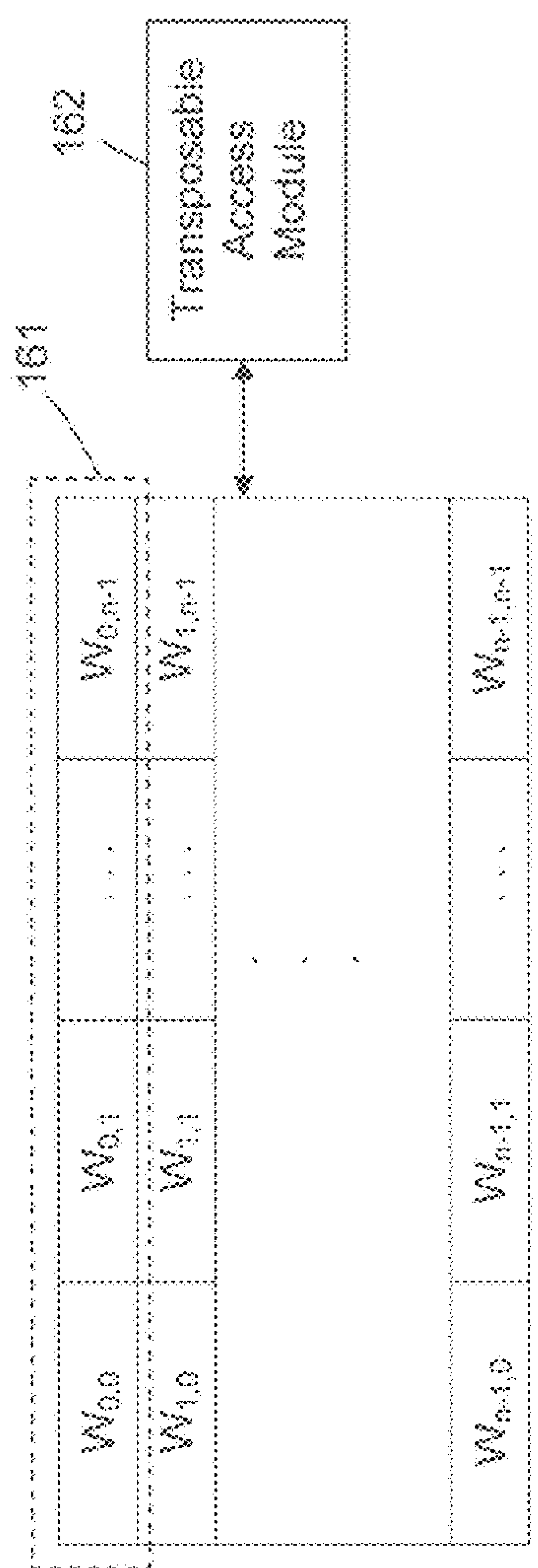
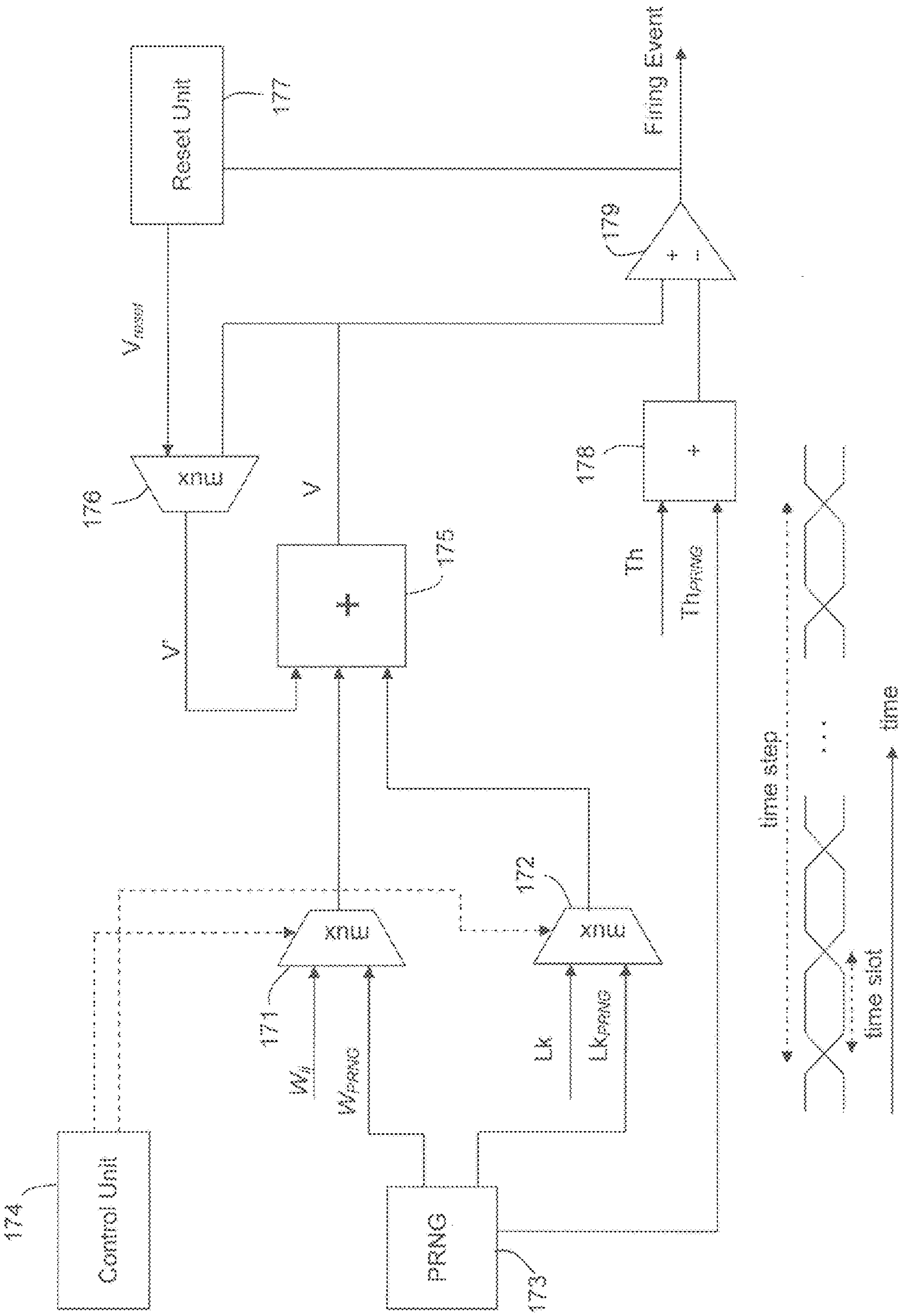


FIG. 7B



Reset Unit
177
Firing Event
179
V_{reset}
176
mux
V
V'
175
+
178
Th
Th_{PRNG}
174
Control Unit
171
mux
W_{ij}
W_{PRNG}
172
mux
Lk
Lk_{PRNG}
PRNG
173
time step
time slot
time

FIG. 8

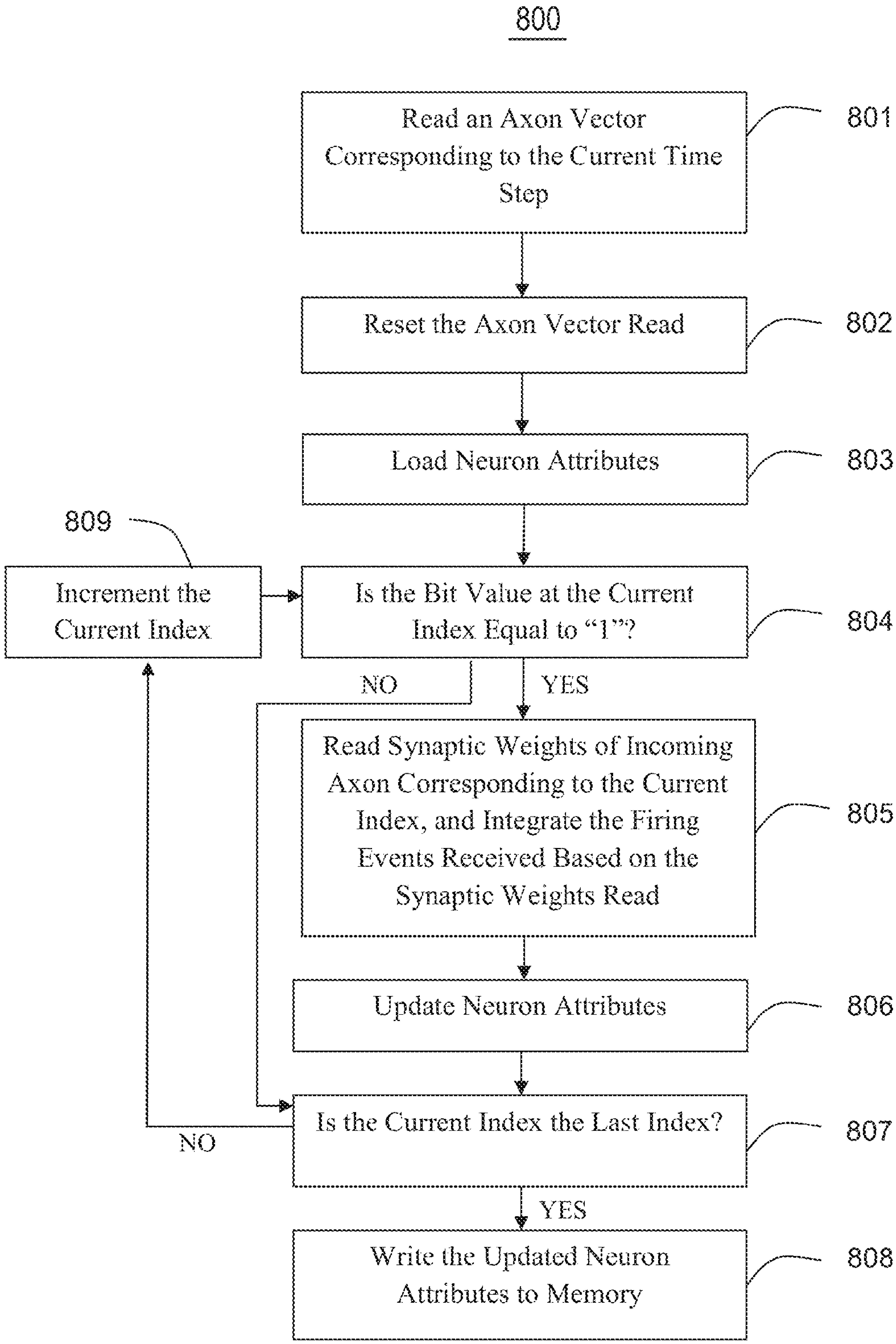

800
Read an Axon Vector Corresponding to the Current Time Step
801
Reset the Axon Vector Read
802
Load Neuron Attributes
803
809
Increment the Current Index
Is the Bit Value at the Current Index Equal to "1"?
804
NO
YES
Read Synaptic Weights of Incoming Axon Corresponding to the Current Index, and Integrate the Firing Events Received Based on the Synaptic Weights Read
805
Update Neuron Attributes
806
Is the Current Index the Last Index?
807
NO
YES
Write the Updated Neuron Attributes to Memory
808

FIG. 9

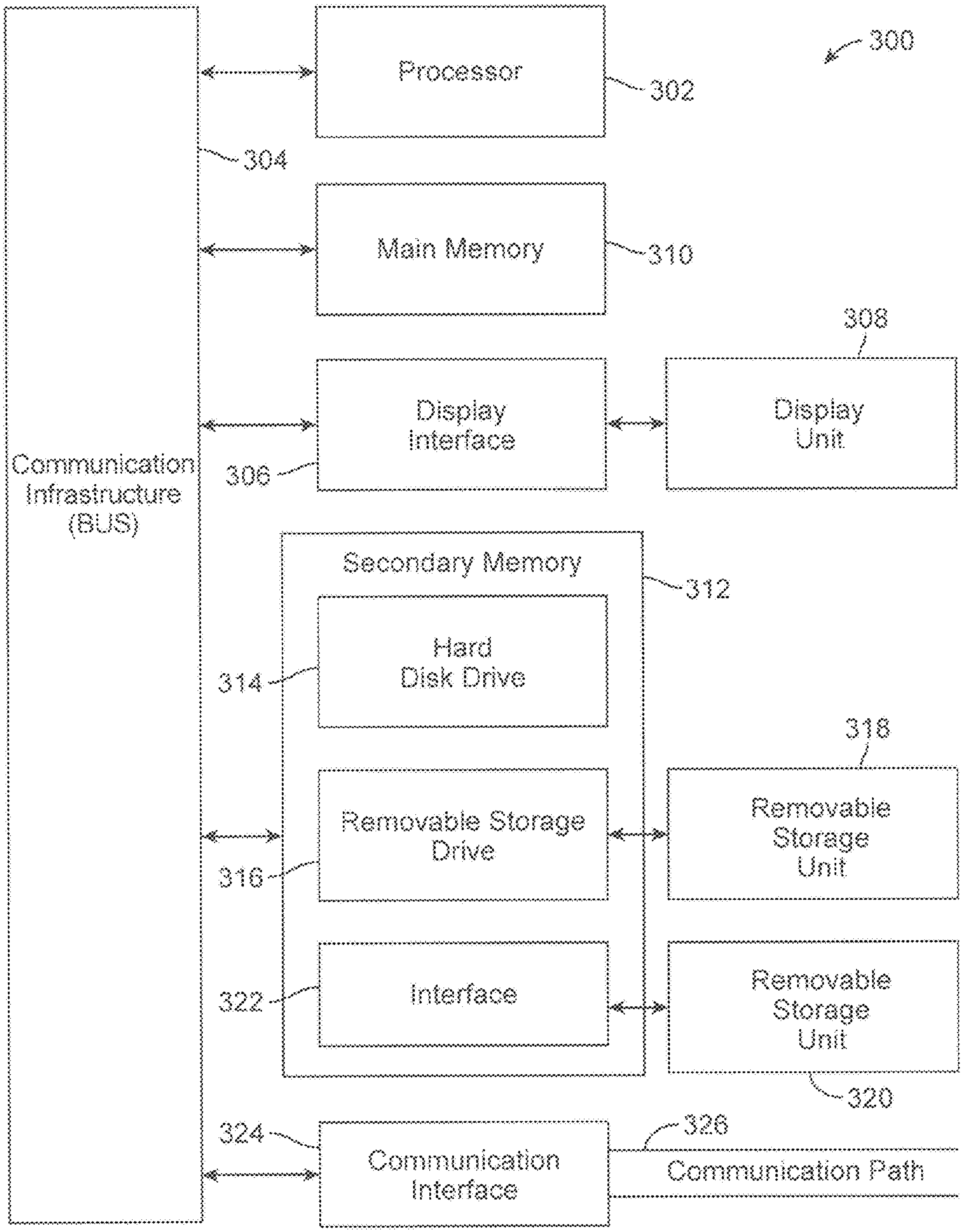

300
Processor
302
304
Main Memory
310
308
Display Interface
Display Unit
306
Communication Infrastructure (BUS)
Secondary Memory
312
Hard Disk Drive
314
318
Removable Storage Drive
Removable Storage Unit
316
Interface
Removable Storage Unit
322
320
324
326
Communication Interface
Communication Path

FIG. 10

TIME-DIVISION MULTIPLEXED NEUROSYNAPTIC MODULE WITH IMPLICIT MEMORY ADDRESSING FOR IMPLEMENTING A UNIVERSAL SUBSTRATE OF ADAPTATION

This invention was made with Government support under HR0011-09-C-0002 awarded by Defense Advanced Research Projects Agency (DARPA). The Government has certain rights in this invention.

BACKGROUND

Embodiments of the invention relate to neuromorphic and synaptronic computation, and in particular, a time-division multiplexed neurosynaptic module with implicit memory addressing for implementing a universal substrate of adaptation.

Neuromorphic and synaptronic computation, also referred to as artificial neural networks, are computational systems that permit electronic systems to essentially function in a manner analogous to that of biological brains. Neuromorphic and synaptronic computation do not generally utilize the traditional digital model of manipulating 0s and 1s. Instead, neuromorphic and synaptronic computation create connections between processing elements that are roughly functionally equivalent to neurons of a biological brain. Neuromorphic and synaptronic computation may comprise various electronic circuits that are modeled on biological neurons.

In biological systems, the point of contact between an axon of a neuron and a dendrite on another neuron is called a synapse, and with respect to the synapse, the two neurons are respectively called pre-synaptic and post-synaptic. The essence of our individual experiences is stored in conductance of the synapses. The synaptic conductance changes with time as a function of the relative spike times of pre-synaptic and post-synaptic neurons, as per spike-timing dependent plasticity (STDP). The STDP rule increases the conductance of a synapse if its post-synaptic neuron fires after its pre-synaptic neuron fires, and decreases the conductance of a synapse if the order of the two firings is reversed.

BRIEF SUMMARY

Embodiments of the invention relate to a time-division multiplexed neurosynaptic module with implicit memory addressing for implementing a universal substrate of adaptation. One embodiment comprises a neurosynaptic device including a memory device that maintains neuron attributes for multiple neurons. The module further includes multiple bit maps that maintain incoming firing events for different periods of delay, and a multi-way processor. The processor includes a memory array that maintains a plurality of synaptic weights. The processor integrates incoming firing events in a time-division multiplexing manner. Incoming firing events are integrated based on the neuron attributes and the synaptic weights maintained.

Another embodiment comprises maintaining neuron attributes for multiple neurons in a memory device, and maintaining incoming firing events for different periods of delay in multiple bit maps. Incoming firing events are integrated in a time-division multiplexing manner using a using a multi-way processor, wherein the processor includes a memory array that maintains a plurality of synaptic weights. The incoming firing events are integrated based on the neuron attributes and the synaptic weights maintained.

These and other features, aspects and advantages of the present invention will become understood with reference to the following description, appended claims and accompanying figures.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

FIG. **1**A illustrates a neurosynaptic core circuit, in accordance with an embodiment of the invention;

FIG. **1**B illustrates an example neural network, in accordance with an embodiment of the invention;

FIG. **2** is a block diagram of a time-division multiplexed neurosynaptic module, in accordance with an embodiment of the invention;

FIG. **3**A illustrates a neuron data memory device, in accordance with an embodiment of the invention;

FIG. **3**B illustrates example neuron attributes maintained in an entry of a neuron data memory device, in accordance with an embodiment of the invention;

FIG. **4**A illustrates a routing data lookup table, in accordance with an embodiment of the invention;

FIG. **4**B illustrates example routing information maintained in an entry of a routing data lookup table, in accordance with an embodiment of the invention;

FIG. **5** illustrates a block diagram of a scheduler device, in accordance with an embodiment of the invention;

FIG. **6** illustrates a collection of axon activity bit maps, in accordance with an embodiment of the invention;

FIG. **7**A illustrates a non-transposable synapse data memory array, in accordance with an embodiment of the invention;

FIG. **7**B illustrates a transposable synapse data memory array, in accordance with an embodiment of the invention;

FIG. **8** illustrates an example computation circuit of a multi-way parallel processor device, in accordance with an embodiment of the invention;

FIG. **9** illustrates a flowchart of an example process for processing incoming firing events in a time-division multiplexed manner, in accordance with an embodiment of the invention; and

FIG. **10** is a high level block diagram showing an information processing system useful for implementing one embodiment of the invention.

DETAILED DESCRIPTION

Embodiments of the invention relate to a time-division multiplexed neurosynaptic module with implicit memory addressing for implementing a universal substrate of adaptation. One embodiment comprises a neurosynaptic device including a memory device that maintains neuron attributes for multiple neurons. The module further includes multiple bit maps that maintain incoming firing events for different periods of delay, and a multi-way processor. The processor includes a memory array that maintains a plurality of synaptic weights. The processor integrates incoming firing events in a time-division multiplexing manner. Incoming firing events are integrated based on the neuron attributes and the synaptic weights maintained.

Another embodiment comprises maintaining neuron attributes for multiple neurons in a memory device, and maintaining incoming firing events for different periods of delay in multiple bit maps. Incoming firing events are integrated in a time-division multiplexing manner using a using a multi-way processor, wherein the processor includes a memory array that maintains a plurality of synaptic weights. The incoming firing events are integrated based on the neuron attributes and the synaptic weights maintained.

Each bit map corresponds to a period of delay. The processor reads and processes a bit map only when a corresponding period of delay for the bit map has elapsed. For each neuron, the neuron attributes for the neuron includes a membrane potential variable and a threshold parameter for the neuron. For each neuron, the processor generates an outgoing firing event when a membrane potential variable of the neuron exceeds a threshold parameter of the neuron. The processor updates the neuron attributes for the neurons based on the integrated incoming firing events, and writes the updated neuron attributes to the memory device. The processor updates the synaptic weights maintained based on a learning rule.

A routing data lookup table maintains routing information for the neurons. A scheduler device is configured to receive firing events, and write each firing event received to a bit map, wherein the firing event will be delivered to a target incoming axon only after a period of delay corresponding to the bit map has elapsed.

Each time step is divided into multiple time slots. In one embodiment, the memory array is a transposable memory array.

In one embodiment, a transposable access module provides transposable access to the synaptic weights maintained, wherein transposable access to the synaptic weights maintained facilitates enhanced synaptic plasticity and learning. Each bit map maintains at least one axon vector. Each axon vector identifies axons that received incoming firing events in a corresponding time step.

For each time step, the processor reads each axon vector corresponding to the time step, and resets each axon vector read. For each axon vector read in each time step, the processor loads neuron attributes from the memory device. For each axon that received an incoming firing event in the time step, the processor reads synaptic weights of synapses interconnecting the axon with the neurons, and updates at least one neuron attribute of each neuron based on a synaptic weight read. For each axon vector read in each time step, the processor writes updated neuron attributes for the neurons to the memory device, and generates an update vector identifying neurons that generated an outgoing firing event in the time step.

The term digital neuron as used herein represents an framework configured to simulate a biological neuron. An digital neuron creates connections between processing elements that are roughly functionally equivalent to neurons of a biological brain. As such, a neuromorphic and synaptronic computation comprising digital neurons according to embodiments of the invention may include various electronic circuits that are modeled on biological neurons. Further, a neuromorphic and synaptronic computation comprising digital neurons according to embodiments of the invention may include various processing elements (including computer simulations) that are modeled on biological neurons. Although certain illustrative embodiments of the invention are described herein using digital neurons comprising digital circuits, the present invention is not limited to digital circuits. A neuromorphic and synaptronic computation according to embodiments of the invention can be implemented as a neuromorphic and synaptronic framework comprising circuitry, and additionally as a computer simulation. Indeed, embodiments of the invention can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements.

FIG. **1**A illustrates a neurosynaptic core circuit **10**, in accordance with an embodiment of the invention. The core circuit **10** is a neural core circuit. The core circuit **10** comprises multiple incoming axons **15** and multiple neurons **11**. Each neuron **11** and each axon **15** has configurable operational parameters. The core circuit **10** further comprises a synaptic crossbar **12** including multiple synapses **31**, multiple rows/axon paths **26**, and multiple columns/dendrite paths **34**.

Each synapse **31** communicates firing events (e.g., spike events) between an axon **15** and a neuron **11**. Specifically, each synapse **31** is located at cross-point junction between an axon path **26** and a dendrite path **34**, such that a connection between the axon path **26** and the dendrite path **34** is made through said synapse **31**. Each axon **15** is connected to an axon path **26**, such that said axon **15** sends spikes to the connected axon path **26**. Each neuron **11** is connected to a dendrite path **34**, such that said neuron **11** receives spikes from the connected dendrite path **34**.

Each synapse **31** has a synaptic weight. The synaptic weights of the synapses **31** of the core circuit **10** may be represented by a weight matrix W, wherein an element W_{ij} of the matrix W represents a synaptic weight of a synapse **31** located at a row/axon path i and a column/dendrite path j of the crossbar **12**. In one embodiment, the synapses **31** are binary memory devices. Each synapse **31** can have a weight "0" indicating that said synapse **31** is non-conducting, or a weight "1" indicating that said synapse **31** is conducting. A learning rule such as spike-timing dependent plasticity (STDP) may be applied to update the synaptic weights of the synapses **31**.

FIG. **1**B illustrates an example neural network **50**, in accordance with an embodiment of the invention. The neural network **50** is a scalable neuromorphic and synaptronic architecture. The neural network **50** includes multiple chip structures **70**. Each chip structure **70** comprises multiple core circuits **10**. An event routing system **75** of the neural network **50** routes firings events between core circuits **10** of the chip structures **70**. A core circuit **10** of the neural network **50** may send firing events to, and receive firing events from, a different core circuit **10** of the same chip structure **70** or a different chip structure **70**.

In one embodiment, the routing system **75** comprises point-to-point connections. In another embodiment, the routing system **75** comprises network-on-chip channels and interchip routers.

In one embodiment, a neural network including at least one core circuit **10** may be implemented as a time-division multiplexed neurosynaptic module. A neurosynaptic module is an electronic device comprising at least one multi-way parallel processor.

FIG. **2** is a block diagram of a time-division multiplexed neurosynaptic module **100**, in accordance with an embodiment of the invention. The neurosynaptic module **100** comprises at least one multi-way parallel processor device ("processor") **150**. Each processor **150** multiplexes computation and control logic for a plurality of neurons **11**. In one embodiment, each processor **150** multiplexes computation and control logic for neurons **11** of one core circuit **10**. In another embodiment, each processor **150** multiplexes computation and control logic for neurons **11** of different core circuits **10**.

The processors **150** of the neurosynaptic module **100** run in parallel. Each processor **150** has a corresponding neuron data memory device **200**, a corresponding collection **251** of axon activity bit maps **250** (FIG. **6**), a corresponding scheduler device ("scheduler") **350**, and a corresponding routing data lookup table (LUT) **400**. A neuron data memory device **200** maintains neuron attributes **215** (FIG. **3**B) for multiple neurons **11**. In one embodiment, the memory device **200** maintains neuron attributes **215** for neurons **11** of one core circuit **10**. In another embodiment, the memory device **200** maintains neuron attributes **215** for neurons **11** of different core circuits **10**. A routing data LUT **400** maintains routing information for multiple neurons **11**. A collection **251** of axon activity bit maps **250** maintains incoming firing events that are delivered to target incoming axons **15** in future time steps. Each bit of a bit map **250** corresponds to an incoming axon **15**.

The neurosynaptic module **100** is connected to an interconnect network **450** that communicates firing events between multiple neurosynaptic modules **100**. In one embodiment, firing events are propagated through the interconnect network **450** in the form of address-event packets. Each address-event packet includes a firing event encoded as a binary address that represents a target incoming axon **15**, a time stamp indicating when the firing event was generated, and a predetermined delivery delay indicating when the firing event should be delivered to the target incoming axon **15**. The scheduler **350** receives address-events from, and sends address-event packets to, the interconnect network **450**.

Each processor **150** comprises a synapse data memory array **160** and a computation logic circuit ("computation circuit") **170**. A memory array **160** maintains synaptic connectivity information for multiple neurons **11**. In one embodiment, a memory array **160** is a transposable memory array including configurable synaptic connectivity information. In another embodiment, a memory array **160** is a non-transposable memory array including static synaptic connectivity information. A computation circuit **170** integrates incoming firing events for a current time step, and updates neuron attributes **215** based on the firing events integrated.

A processor **150** that multiplexes computation and control logic for n neurons **11** is an n-way processor **150**, wherein n is an integer value. The computation circuit **170** of an n-way processor **150** is time-multiplexed n times.

The total number of neurons **11** represented by the neurosynaptic module **100** is equal to the product of the number of processors **150** contained within the neurosynaptic module **100**, and the number of times each processor **150** of the neurosynaptic module **100** is time-multiplexed. For example, if the neurosynaptic module **100** contains Y processors **150** and each processor **150** is time-multiplexed n times, the total number of neurons **11** represented by the neurosynaptic module **100** is Y×n, where Y and n are positive integer values.

The optimal number of neurons **11** that a neurosynaptic module **100** may represent is dependent on several factors, including the connectivity of the neurons **11**, communication power overhead, and the performance of the synapse data memory array **160** of each processor **150**.

FIG. **3**A illustrates a neuron data memory device **200**, in accordance with an embodiment of the invention. As stated above, each processor **150** has a corresponding neuron data memory device **200** that maintains neuron attributes **215** for multiple neurons **11**. The memory device **200** comprises multiple entries **211**. Each entry **211** maintains neuron attributes **215** for a corresponding neuron **11**.

As shown in FIG. **3**A, the memory device **200** maintains neuron attributes **215** for neurons Neuron 0, Neuron 1, . . . , and Neuron n−1, wherein n represents the number of neurons **11** that the memory device **200** maintains information for.

FIG. **3**B illustrates example neuron attributes **215** maintained in an entry **211** of a neuron data memory device **200**, in accordance with an embodiment of the invention. In one embodiment, each entry **211** maintains the following neuron attributes **215** for a corresponding neuron **11**: a membrane potential variable (V), a threshold parameter (Th), a leak rate parameter (Lk), and synaptic excitation/inhibition strengths for each possible axon type (Syn0, Syn1, Syn2, etc.).

FIG. **4**A illustrates a routing data lookup table **400**, in accordance with an embodiment of the invention. As stated above, each processor **150** has a corresponding routing data LUT **400** that maintains routing information for multiple neurons **11**. The LUT **400** comprises multiple entries **411**. Each entry **411** maintains routing information for a corresponding neuron **11**.

As shown in FIG. **4**A, the LUT **400** maintains routing information for neurons Neuron 0, Neuron 1, . . . , and Neuron n−1, wherein n represents the number of neurons **11** that the LUT **400** maintains information for.

FIG. **4**B illustrates example routing information maintained in an entry **411** of a routing data lookup table **400**, in accordance with an embodiment of the invention. In one embodiment, each entry **411** maintains the following routing information for a corresponding neuron **11**: fanout (F) and delivery delay (ΔT). The fanout of a neuron **11** indicates a target incoming axon **15** that the neuron **11** sends outgoing firing events to. The delivery delay of a neuron **11** indicates when an outgoing firing event generated by the neuron **11** should be delivered to a target incoming axon **15** for processing.

FIG. **5** illustrates a block diagram of a scheduler device **350**, in accordance with an embodiment of the invention. As stated above, each processor **150** has a corresponding scheduler device **350**. The scheduler **350** comprises a controller **351**, an off-module buffer ("buffer") **352**, a decoder unit ("decoder") **353**, and an encoder unit ("encoder") **354**.

The controller **351** generates time steps that triggers when a corresponding processor **150** integrates incoming firing events.

The decoder **353** receives from the interconnect network **450** (i.e., off-module) incoming address events packets that include firing events generated by other neurosynaptic modules **100**. The decoder **353** decodes each incoming address event packet received. In one embodiment, decoded incoming firing events are temporarily held in the buffer **352** before the controller **351** copies the firing events to an axon activity bit map **250**. The buffer **352** is cleared after the controller **351** has copied the decoded incoming firing events to a bit map **250**.

The controller **351** generates axon vectors **255** (FIG. **6**). Each axon vector **255** corresponds to a time step (i.e., a current time step or a future time step). Each axon vector **255** represents axon activity for incoming axons **15** in a corresponding time step. Each index of an axon vector **255** corresponds to an incoming axon **15**. In one embodiment, each index with a bit value of "1" indicates that a corresponding axon **15** received a firing event. Each index with a bit value of "0" indicates that a corresponding axon **15** did not receive a firing event. In one embodiment, each axon vector **255** represents axon activity for incoming axons **15** of a corresponding core circuit **10** in a corresponding time step.

The controller **351** writes each axon vector **255** generated to an axon activity bit map **250** of the collection **251**, wherein the bit map **250** corresponds to the same time step that said axon vector **255** corresponds to.

In one embodiment, for each incoming firing event, the controller **351** computes the difference d between the arrival time of said firing event at the scheduler **350** and the time stamp indicating when said firing event was generated. If the difference d is less than a predetermined delivery delay x, the firing event is maintained in a bit map **250** for a delay period D equal to the difference between x and d to achieve x time steps from firing event generation to firing event delivery. The processor **150** reads the firing event from the bit map **250** at the end of the delay period.

For example, if the delivery delay for a firing event is 9 time steps and the firing event arrives at the scheduler **350** within 3 time steps from generation, the scheduler **350** delays the delivery of the firing event by 6 time steps, such that the processor **150** reads the firing event from a bit map **250** only at the end of 9 time steps from generation.

In each time step, the scheduler **350** receives an update vector **257** from a corresponding processor **150**, wherein the update vector **257** represents firing activity of multiple neurons **11** during said time step. Each index of an update vector **257** corresponds to a neuron **11**. Each index with a bit value of "1" indicates that a corresponding neuron **11** generated an outgoing firing event. Each index with a bit value of "0" indicates that a corresponding neuron **11** did not generate an outgoing firing event.

Each outgoing firing event targets either an incoming axon **15** of the same neurosynaptic module **100** (i.e., on-module) or a different neurosynaptic module **100** (i.e., off-module). For each index of an update vector **257** with a bit value of "1", the controller **351** looks up routing information for a corresponding neuron **11** in the LUT **400**. If the target axon **15** for an outgoing firing event is on-module (i.e., on the same neurosynaptic module **100**), the controller **351** determines, based on the current time step and the delivery delay of the firing event, which bit map **250** of the collection **251** to update, and updates a bit of the determined bit map **250** accordingly. If the target axon **15** for an outgoing firing event is off-module (i.e., on a different neurosynaptic module **100**), the encoder **354** encapsulates the outgoing firing event as an outgoing address event packet, and sends the outgoing address event packet to the interconnect network **450**.

FIG. **6** illustrates a collection **251** of axon activity bit maps **250**, in accordance with an embodiment of the invention. As stated above, each processor **150** has a corresponding collection **251** of axon activity bit maps **250**. Each bit map **250** maintains at least one axon vector **255**.

Each bit map **250** of the collection **251** corresponds to a future time step. Specifically, each bit map **250** corresponds to a duration of delay. For example, as shown in FIG. **6**, the collection **251** maintains bit maps **250** corresponding to delays ranging from one time step to sixteen time steps from the current time step t. A first bit map **250** corresponds to axon activity that will occur after a delay of one time step, a second bit map **250** corresponds to axon activity after a delay of two time steps, . . . , and a sixteenth bit map **250** corresponds to axon activity after a delay of sixteen time steps. Each bit map **250** maintains one or more axon vectors **255**, wherein each axon vector **255** indicates the axon activity of incoming axons **15** in a time step equal to the current time step t plus a corresponding delay of said bit map **250**.

A corresponding processor **150** iterates through each bit map **255** of the collection **251**. Specifically, the processor **105** reads an axon vector **255** from a bit map **250** only when a delay corresponding to said bit map **250** has elapsed. For example, in time step t+1, the processor **150** reads axon vectors **255** from the first bit map **250** corresponding to time step t+1. In time step t+16, the processor **150** reads axon vectors **255** from the sixteenth bit map **250** corresponding to time step t+16.

Each axon vector **255** is reset after said axon vector **255** has been read by the corresponding processor **150**. After each axon vector **255** of the sixteenth bit map **250** has been read, the processor **150** begins another iteration through each bit map **250** of the collection **251**. For example, in time step t+17, the processor **150** reads axon vectors from the first bit map **250**.

FIG. **7**A illustrates a non-transposable synapse data memory array **160**, in accordance with an embodiment of the invention. As stated above, each processor **150** has a synapse data memory array **160**. In one embodiment, the memory array **160** is a non-transposable memory array that maintains static synaptic connectivity information for multiple neurons **11**.

The memory array **160** comprises multiple entries **161**. Each entry **161** maintains synaptic weights for a corresponding neuron **11**. As shown in FIG. **7**A, a first entry **161** includes synaptic weights $W_{0,0}$, $W_{0,1}$, . . . , and $W_{0,n-1}$.

FIG. **7**B illustrates a transposable synapse data memory array **160**, in accordance with an embodiment of the invention. In another embodiment, the memory array **160** of a processor **150** is a transposable memory array maintaining configurable synaptic connectivity information for multiple neurons **11**. The memory array **160** has a corresponding transposable access module **162** that facilitates transposable access to the memory array **160**. Synaptic weights may be read from, and written to, the memory array **160**, in both horizontal and vertical directions for enhanced learning operation. The synaptic weights maintained may be updated based on a learning rule, and/or the firing activity of a corresponding neuron **11**.

The memory array **160** comprises multiple entries **161**. Each entry **161** maintains synaptic weights for a corresponding neuron **11**. As shown in FIG. **7**A, a first entry **161** includes synaptic weights $W_{0,0}$, $W_{0,1}$, . . . , and $W_{0,n-1}$.

FIG. **8** illustrates an example computation circuit **170** of a multi-way parallel processor device **150**, in accordance with an embodiment of the invention. As stated above, each processor **150** has a computation circuit **170**. In one embodiment, the circuit **170** comprises a first multiplexer **171**, a second multiplexer **172**, a pseudo-random number generator (PRNG) **173**, a time-division multiplexing control unit **174**, a first adder unit ("first adder") **175**, a third multiplexer **176**, a reset unit **177**, a second adder unit ("second adder") **178**, and a comparator unit ("comparator") **179**.

To implement an n-way processor **150**, the computation circuit **170** is time-multiplexed n times, wherein n represents the number of neurons **11** that the processor multiplexes computation and control logic for. The control unit **174** divides each time step into n time slots. In each time slot, incoming firing events targeting a corresponding incoming axon are integrated. The control unit **174** is further configured to send control signals to components of the circuit **170**.

The PRNG **173** generates random numbers for use in stochastic operations. For example, the PRNG **173** may be used to generate a random synaptic weight W_{PRNG}, a random leak rate Lk_{PRNG}, and a random threshold Th_{PRNG}.

At the beginning of each time step, the processor **150** reads an axon vector **255** from a bit map **250** corresponding to said time step. The axon vector **255** is reset after it is read by the processor **150**. The processor **150** is loaded with neuron attributes for all neurons **11** that a corresponding memory device **200** maintains information for. In one example implementation, the neuron attributes are loaded into local registers (e.g., latches or flip-flops) of the processor **150**.

The processor **150** iterates through each index of the axon vector **250**. For each index i of the axon vector **255** read with a bit value of "1", each synaptic weight maintained in the i^{th} entry of the memory array **160** is read. Each synaptic weight W_{ij} that is read from the i^{th} entry of the memory array **160** is conveyed to corresponding computation circuit **170**, and the first multiplexer **171** selects between the synaptic weight W_{ij} and a random synaptic weight W_{PRNG}.

For the first addition that corresponds to the first index of the axon vector **255** with a bit value of "1", the first adder **175** increments the membrane potential variable V (loaded from the i^{th} entry of the corresponding memory device **200**) by the value selected by the first multiplexer **171**. For subsequent additions (i.e., the remaining indices of the axon vector **255** with a bit value of "1"), the first adder **175** increments a modified membrane potential variable V' by the value selected by the first multiplexer **171**. The modified membrane potential variable V' is a temporary variable provided by the third multiplexer **176**. The third multiplexer **176** selects between an updated membrane potential variable V provided by the first adder **175** and a reset membrane potential variable V_{reset} generated by the reset unit **177**.

The second multiplexer **172** selects between a leak rate parameter Lk (loaded from the i^{th} entry of the corresponding memory device **200**) and a random leak rate Lk_{PRNG}. After each synaptic weight W_{ij} has been read from the i^{th} entry of the memory array **160**, the first adder **175** increments the modified membrane potential variable V' by the value selected by the second multiplexer **172**.

The second adder **178** increments the threshold parameter Th (loaded from the i^{th} entry of the corresponding memory device **200**) by a random threshold Th_{PRNG}. In another embodiment, the unit **178** is a multiplexer. The comparator **179** generates a firing event if the comparator **179** determines that the updated membrane potential variable V has exceeded the value provided by the second adder **178**. After the firing event is generated, the membrane potential variable V is reset to V_{reset}.

When the processor **150** has finished iterating through each index of the axon vector **255**, the updated neuron attributes **215** (e.g., the updated membrane potential variable V) are written to the memory device **200**. An update vector **257** representing the firing activity of neurons **11** in said time step is generated and sent to the scheduler **350**.

FIG. **9** illustrates a flowchart of an example process **800** for processing incoming firing events in a time-division multiplexed manner, in accordance with an embodiment of the invention. In process block **801**, read an axon vector corresponding to the current time step. In process block **802**, reset the axon vector read. In process block **803**, load neuron attributes. In process block **804**, determine if the bit value at the current index of the axon vector is "1". If the bit value is 1, proceed to process block **805**. If the bit value is not "1", proceed to process block **807**.

In process block **805**, read synaptic weights of the incoming axon corresponding to the current index and integrate the firing events received based on the synaptic weights read. In process block **806**, update neuron attributes. In process block **807**, determine whether the current index is the last index of the axon vector. If the current index is the last index, proceed to process block **808**. If the current index is not the last index, proceed to process block **809**. In process block **808**, write the updated neuron attributes to memory. In process block **809**, increment the current index. Process blocks **803-809** are repeated for each neuron.

FIG. **10** is a high level block diagram showing an information processing system **300** useful for implementing one embodiment of the invention. The computer system includes one or more processors, such as processor **302**. The processor **302** is connected to a communication infrastructure **304** (e.g., a communications bus, cross-over bar, or network).

The computer system can include a display interface **306** that forwards graphics, text, and other data from the communication infrastructure **304** (or from a frame buffer not shown) for display on a display unit **308**. The computer system also includes a main memory **310**, preferably random access memory (RAM), and may also include a secondary memory **312**. The secondary memory **312** may include, for example, a hard disk drive **314** and/or a removable storage drive **316**, representing, for example, a floppy disk drive, a magnetic tape drive, or an optical disk drive. The removable storage drive **316** reads from and/or writes to a removable storage unit **318** in a manner well known to those having ordinary skill in the art. Removable storage unit **318** represents, for example, a floppy disk, a compact disc, a magnetic tape, or an optical disk, etc. which is read by and written to by removable storage drive **316**. As will be appreciated, the removable storage unit **318** includes a computer readable medium having stored therein computer software and/or data.

In alternative embodiments, the secondary memory **312** may include other similar means for allowing computer programs or other instructions to be loaded into the computer system. Such means may include, for example, a removable storage unit **320** and an interface **322**. Examples of such means may include a program package and package interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units **320** and interfaces **322**, which allows software and data to be transferred from the removable storage unit **320** to the computer system.

The computer system may also include a communication interface **324**. Communication interface **324** allows software and data to be transferred between the computer system and external devices. Examples of communication interface **324** may include a modem, a network interface (such as an Ethernet card), a communication port, or a PCMCIA slot and card, etc. Software and data transferred via communication interface **324** are in the form of signals which may be, for example, electronic, electromagnetic, optical, or other signals capable of being received by communication interface **324**. These signals are provided to communication interface **324** via a communication path (i.e., channel) **326**. This communication path **326** carries signals and may be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link, and/or other communication channels.

In this document, the terms "computer program medium," "computer usable medium," and "computer readable medium" are used to generally refer to media such as main memory **310** and secondary memory **312**, removable storage drive **316**, and a hard disk installed in hard disk drive **314**.

Computer programs (also called computer control logic) are stored in main memory **310** and/or secondary memory **312**. Computer programs may also be received via communication interface **324**. Such computer programs, when run, enable the computer system to perform the features of the present invention as discussed herein. In particular, the computer programs, when run, enable the processor **302** to perform the features of the computer system. Accordingly, such computer programs represent controllers of the computer system.

From the above description, it can be seen that the present invention provides a system, computer program product, and method for implementing the embodiments of the invention. The present invention further provides a non-transitory computer-useable storage medium for hierarchical routing and two-way information flow with structural plasticity in neural networks. The non-transitory computer-useable storage medium has a computer-readable program, wherein the program upon being processed on a computer causes the computer to implement the steps of the present invention according to the embodiments described herein. References in the claims to an element in the singular is not intended to mean "one and only" unless explicitly so stated, but rather "one or more." All structural and functional equivalents to the elements of the above-described exemplary embodiment that are currently known or later come to be known to those of ordinary skill in the art are intended to be encompassed by the present claims. No claim element herein is to be construed under the provisions of 35 U.S.C. section 112, sixth paragraph, unless the element is expressly recited using the phrase "means for" or "step for."

The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

The corresponding structures, materials, acts, and equivalents of all means or step plus function elements in the claims below are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed. The description of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the invention. The embodiment was chosen and described in order to best explain the principles of the invention and the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A neurosynaptic device, comprising:
- a memory device that maintains neuron attributes for multiple neurons;
- multiple bit maps, wherein each bit map corresponds to a period of delay of different periods of delay, the bit map comprises at least one axon vector, each index of the at least one axon vector corresponds to an incoming axon of multiple incoming axons, and the index comprises a bit value indicating whether the corresponding incoming axon receives an incoming firing event in a future time step that occurs after the corresponding period of delay has elapsed; and
- a multi-way processor including a memory array that maintains a plurality of synaptic weights, wherein the multi-way processor is configured to:
  - for each period of delay of the different periods of delay:
    - in response to elapse of the period of delay, integrate one or more incoming firing events in accordance with a bit map corresponding to the period of delay in a time-division multiplexing manner, wherein the incoming firing events are integrated based on the neuron attributes and the synaptic weights; and
    - generate an update vector representing firing activity of the multiple neurons based on the integrated incoming firing events, wherein at least one of the multiple bit maps is updated based on the update vector.

2. The neurosynaptic device of claim **1**, wherein:
- the update vector identifies which of the multiple neurons generated an outgoing firing event; and
- the multi-way processor is further configured to:
  - for each period of delay of the different periods of delay:
    - in response to elapse of the period of delay, read and process a bit map corresponding to the period of delay.

3. The neurosynaptic device of claim **2**, wherein:
- for each neuron, the neuron attributes include a membrane potential variable for said neuron and a threshold parameter for said neuron.

4. The neurosynaptic device of claim **3**, wherein:
- the multi-way processor is further configured to:
  - for each neuron:
    - determine whether a membrane potential variable for said neuron exceeds a threshold parameter for said neuron;
    - generate an outgoing firing event in response to determining the membrane potential variable for said neuron exceeds the threshold parameter for said neuron, wherein at least one of the multiple bit maps is updated based on the outgoing firing event.

5. The neurosynaptic device of claim **4**, wherein the multi-way processor:
- updates the neuron attributes for the multiple neurons based on the integrated incoming firing events; and
- writes the updated neuron attributes to the memory device.

6. The neurosynaptic device of claim **5**, further comprising:
- a routing data lookup table that maintains routing information for the multiple neurons.

7. The neurosynaptic device of claim **6**, further comprising:
- a scheduler device configured to:
  - receive one or more firing events; and
  - for each firing event received, update a bit map of the multiple bit maps by updating an index of an axon vector of the bit map with a bit value indicating that a corresponding incoming axon receives the firing event in a future time step that occurs after a corresponding period of delay has elapsed, wherein the firing event is delivered to the corresponding incoming axon in the future time step.

8. The neurosynaptic device of claim **7**, wherein:
- the multi-way processor is further configured to update the synaptic weights based on a learning rule.

9. The neurosynaptic device of claim **8**, wherein:
- each period of delay of the different periods of delay corresponds to a future time step that occurs after the period of delay has elapsed; and
- each time step is divided into multiple time slots.

10. The neurosynaptic device of claim **1**, wherein:
- the memory array is a transposable memory array.

11. The neurosynaptic device of claim **1**, further comprising:
- a transposable access module for transposable access to the synaptic weights, wherein the transposable access to the synaptic weights facilitates enhanced synaptic plasticity and learning.

12. The neurosynaptic device of claim **11**, wherein:
- each bit map maintains at least one axon vector; and
- each axon vector identifies axons that receive incoming firing events in a corresponding time step.

13. The neurosynaptic device of claim **12**, wherein the multi-way processor is further configured to:
- for each time step:
  - read each axon vector corresponding to the time step; and
  - reset each axon vector read.

14. The neurosynaptic device of claim 13, wherein the multi-way processor is further configured to:

for each axon vector read in each time step:

load one or more of the neuron attributes from the memory device; and

for each axon that receives an incoming firing event in the time step:

read synaptic weights of synapses interconnecting the axon with the multiple neurons; and

update at least one neuron attribute of each neuron based on a synaptic weight read.

15. The neurosynaptic device of claim 14, wherein the multi-way processor is further configured to:

for each axon vector read in each time step:

write one or more updated neuron attributes for the multiple neurons to the memory device.

16. A method, comprising:

maintaining neuron attributes for multiple neurons in a memory device;

maintaining multiple bit maps, wherein each bit map corresponds to a period of delay of different periods of delay, the bit map comprises at least one axon vector, each index of the at least one axon vector corresponds to an incoming axon of multiple incoming axons, and the index comprises a bit value indicating whether the corresponding incoming axon receives an incoming firing event in a future time step that occurs after the corresponding period of delay has elapsed; and

for each period of delay of the different periods of delay:

in response to elapse of the period of delay, integrating one or more incoming firing events in accordance with a bit map corresponding to the period of delay in a time-division multiplexing manner using a multi-way processor, wherein the multi-way processor includes a memory array that maintains a plurality of synaptic weights, and the incoming firing events are integrated based on the neuron attributes and the synaptic weights; and

generating an update vector representing firing activity of the multiple neurons based on the integrated incoming firing events, wherein at least one of the multiple bit maps is updated based on the update vector.

17. The method of claim 16, wherein:

the update vector identifies which of the multiple neurons generated an outgoing firing event.

18. The method of claim 17, further comprising:

for each period of delay of the different periods of delay:

in response to elapse of the period of delay, the multi-way processor reading and processing a bit map corresponding to the period of delay.

19. The method of claim 18, wherein:

for each neuron, the neuron attributes include a membrane potential variable for said neuron and a threshold parameter for said neuron.

20. The method of claim 19, further comprising:

for each neuron:

the multi-way processor determining whether a membrane potential variable for said neuron exceeds a threshold parameter for said neuron; and

the multi-way processor generating an outgoing firing event in response to determining the membrane potential variable for said neuron exceeds the threshold parameter for said neuron, wherein at least one of the multiple bit maps is updated based on the outgoing firing event.

21. The method of claim 20, further comprising:

updating the neuron attributes for the multiple neurons based on the integrated incoming firing events; and

writing the updated neuron attributes to the memory device.

22. The method of claim 21, further comprising:

maintaining routing information for the multiple neurons in a routing data lookup table.

23. The method of claim 22, further comprising:

receiving one or more firing events; and

for each firing event received, updating a bit map of the multiple bit maps by updating an index of an axon vector of the bit map with a bit value indicating that a corresponding incoming axon receives the firing event in a future time step that occurs after a corresponding period of delay has elapsed, wherein the firing event is delivered to the corresponding incoming axon in the future time step.

24. The method of claim 16, further comprising:

using a transposable access module for transposable access to the synaptic weights, wherein the transposable access to the synaptic weights facilitates enhanced synaptic plasticity and learning.

25. The method of claim 24, further comprising:

maintaining at least one axon vector, wherein each axon vector identifies axons that receive incoming firing events in a corresponding time step.

26. The method of claim 25, further comprising:

for each time step:

reading each axon vector corresponding to the time step; and

resetting each axon vector read.

27. The method of claim 26, further comprising:

for each axon vector read in each time step:

loading one or more of the neuron attributes from the memory device; and

for each axon that receives an incoming firing event in the time step:

reading synaptic weights of synapses interconnecting the axon with the multiple neurons; and

updating at least one neuron attribute of each neuron based on a synaptic weight read.

28. The method of claim 27, further comprising:

for each axon vector read in each time step:

writing one or more updated neuron attributes for the multiple neurons to the memory device.

29. A non-transitory computer program product comprising a computer-readable hardware storage medium having program code embodied therewith, the program code being executable by a computer to:

maintain neuron attributes for multiple neurons in a memory device;

maintain multiple bit maps, wherein each bit map corresponds to a period of delay of different periods of delay, the bit map comprises at least one axon vector, each index of the at least one axon vector corresponds to an incoming axon of multiple incoming axons, and the index comprises a bit value indicating whether the corresponding incoming axon receives an incoming firing event in a future time step that occurs after the corresponding period of delay has elapsed; and

for each period of delay of the different periods of delay:

in response to elapse of the period of delay integrate one or more incoming firing events in accordance with a bit map corresponding to the period of delay in a time-division multiplexing manner using a multi-way processor, wherein the multi-way processor includes a memory array that maintains a plurality of synaptic weights, and the incoming firing events are integrated based on the neuron attributes and the synaptic weights; and

generate an update vector representing firing activity of the multiple neurons based on the integrated incoming firing events, wherein at least one of the multiple bit maps is updated based on the update vector.

30. The computer program product of claim **29**, the program code further executable by the computer to:

receive one or more firing events; and

for each firing event received, update a bit map of the multiple bit maps by updating an index of an axon vector of the bit map with a bit value indicating that a corresponding incoming axon receives the firing event in a future time step that occurs after a corresponding period of delay has elapsed, wherein the firing event is delivered to the corresponding incoming axon in the future time step.

31. The computer program product of claim **30**, the program code further executable by the computer to:

for each time step:

read an axon vector corresponding to the time step, wherein the axon vector identifies axons that receive incoming firing events in the time step;

reset the axon vector read;

load one or more of the neuron attributes from the memory device;

for each axon that receives an incoming firing event in the time step:

read synaptic weights of synapses interconnecting the axon with the multiple neurons; and

update at least one neuron attribute of each neuron based on a synaptic weight read; and

write one or more updated neuron attributes for the multiple neurons to the memory device.

* * * * *